



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/925,888	08/14/2001	Tim Wilkinson	TRAS-560	3640

7590 06/16/2004
Andrew V. Smith
Sierra Patent Group
P.O. Box 6149
Stateline, NV 89449

EXAMINER

VO, TED T

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 06/16/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/925,888

Applicant(s)

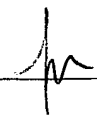
WILKINSON ET AL.

Examiner

Ted T. Vo

Art Unit

2122



-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 14 August 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the application filed on 08/14/2001.

Claims 1-20 are pending in the application.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-20 are rejected under 35 U.S.C. 102(b) as being anticipated by Bacon et al., "Thin Locks: Featherweight Synchronization for Java", ACM 1998.

Given the broadest reasonable interpretation of followed claims in light of the specification:

As per Claim 1: Bacon discloses,

A software program for providing instructions to one or more processors to execute processes on an embedded computing device configured for establishing a network connection with at least one other computing device, comprising:

- (a) an operating system layer (See page 258, right column, first paragraph, AIX);*
- (b) an application framework (See page 258, JDK 1.1.2, JVM, etc.); and*
- (c) a programming environment including a contention locking scheme for setting light object locks (See page 259, Figure 1, Thin Locks), which are handled in user space (referring Figure 1(a)), and heavy object locks (See page 260, Figure 2, Inflated Locks), which are handled at the system level (referring to locked by thread B), and wherein the contention locking scheme is configured to set a light*

Art Unit: 2122

object lock on an initially unlocked object (Figure 1(c), '0' for unlock, Figure 1(d), 'A' for locked by thread A) when a first thread attempts to lock the object, and to maintain a light lock on the object (See Figures 1 and 2, using a field with either 0 for thin lock or 1 for inflated lock) when a nested intra-thread lock is attempted by the first thread (See Figure 1, Count field, or Figure 2, the value '1' or '0' beneath B).

As per Claim 2: Bacon discloses, *The software program of claim 1, wherein the contention locking scheme is further configured to set a heavy object lock on the object when a second thread attempts to lock the object while the object is lightly locked by the lock attempt by the first thread (See Figures 1-2).*

As per Claim 3: Regarding claimed limitation, *The software program of claim 2, wherein the contention locking scheme includes a stack-based local lock slot structure for addressing stack variables to identify threads*, Bacon discloses the structures shown in Figure 1 and Figure 2 having the means of a *stack-based local lock slot structure*. Figure 1, in the field with 'A', it identifies thread A, in figure 2, in the field with 'B', it identifies thread B. Furthermore, "JVM" (Page 258), "thread", or "swap" (Page 260, section 2.3.2) has means of stack base.

As per Claim 4: Bacon discloses, *"The software program of claim 3, wherein a first stack corresponds to a data area of the first thread and a second stack corresponds to a data area of the second thread, and wherein the first and second stacks are separated by at least a reserved area which is set at the end of each stack"*, because field A and field B (referred as stack-based local lock slot structure for addressing stack variables to identify threads) are separated fields.

As per Claim 5: Bacon discloses, *The software program of claim 4, wherein the contention locking scheme is configured to maintain the light lock when an address difference between a current lock slot of the first thread for the lightly locked object and that of the nested intra-thread locking attempt is determined to be less than the reserved area (See Figure 1).*

As per Claim 6: Bacon discloses, *The software program of claim 4, wherein the contention locking scheme is configured to set the heavy lock when an address difference between a current lock slot of the first thread for the lightly locked object and that of the locking attempt by the second thread is determined to be greater than the reserved area (See Figure 2).*

As per Claim 7: Bacon discloses, *The software program of claim 3, wherein the contention locking scheme further includes a lock structure and a lock structure reference in an object header of the object, the lock structure including a lock holder and wait queues (See page 259, Figure 1(b), and left column, section 2, 5 locking scenarios).*

As per Claim 8: Regarding claimed limitation, *The software program of claim 1, wherein the contention locking scheme includes a stack-based local lock slot structure for addressing stack variables to identify threads, Bacon discloses the structures shown in Figure 1 and Figure 2 having the means of a stack-based local lock slot structure. Figure 1, in the field with 'A', it identifies thread A, in figure 2, in the field with 'B', it identifies thread B. Furthermore, "JVM" (Page 258), "thread", or "swap" (Page 260, section 2.3.2) has means of stack base.*

As per Claim 9: Bacon discloses, *The software program of claim 8, wherein the contention locking scheme is configured to maintain the light lock (See Figure 1(b) or 2(a), value '0' or '1' of the first bit) when an address difference between a current lock slot of the first thread for the lightly locked object (See Figure 1(b) or 2(b), 'A' or 'B') and that of the nested intra-thread locking attempt is determined to be less than a reserved area which is set at the end of each stack (See Figure 1(b) or 2(b), Count field).*

As per Claim 10: Regarding limitations:

A software program for providing instructions to one or more processors to execute processes on an embedded computing device configured for establishing a network connection with at least one other computing device, comprising: (a) an operating system layer; (b) an application framework; and (c) a programming environment including a contention locking scheme for setting light object locks, which are handled in user space, and heavy object locks, which are handled at the system level, and wherein the contention locking scheme is configured to set a light object lock on an initially unlocked object when a first thread attempts to lock the object, and to maintain a light lock on the object when a nested intra-thread lock is attempted by the first thread, (See rationale in Claim 1 above) and
(d) wherein the contention locking scheme includes a stack-based local lock slot structure for addressing stack variables to identify threads (See rationale in Claim 3 above), and

Art Unit: 2122

(e) wherein the contention locking scheme further includes a lock structure and a lock structure reference in an object header of the object, the lock structure including a lock holder and wait queues (See rationale in Claim 7 above).

As per Claim 11: Regarding claimed limitation: *A method for executing processes with contention-locking which uses light locks, which are handled at the user level, and heavy locks, which are handled at the system level, on an embedded computing device configured for establishing a network connection with at least one other computing device, comprising the steps of:*

(a) setting a light object lock, for handling at the user level, on an initially unlocked object when a first thread attempts to lock the object; and

(b) determining to maintain the light object lock when a nested intra-thread lock is attempted by the first thread: Claimed limitation has the functionality corresponding to Claim 1. See rationale in Claim 1, step c, above.

As per Claim 12: Regarding claimed limitation: *The method of claim 11, further comprising the step of setting a heavy object lock on the object when a second thread attempts to lock the object while the object is lightly locked by the lock attempt by the first thread:* Claimed limitation has the functionality corresponding to Claim 2. See rationale in Claim 2 above.

As per Claim 13: Regarding claimed limitation: *The method of claim 12, further comprising the step of addressing stack variables to identify threads using a stack-based local lock slot structure:* Claimed limitation has the functionality corresponding to Claim 3. See rationale in Claim 3 above.

As per Claim 14: Regarding claimed limitation: *The method of claim 13, wherein the addressing step includes addressing the first thread at a first stack and addressing the second thread at a second stack, and wherein the first and second stacks are separated by at least a reserved area which is set at the end of each stack:* Claimed limitation has the functionality corresponding to Claim 4. See rationale in Claim 4 above.

As per Claim 15: Regarding claimed limitation: *The method of claim 14, wherein the step of determining to maintain the light object lock includes determining that an address difference between a current lock slot of the first thread for the lightly locked object and that of the nested intra-thread locking attempt is*

less than the reserved area: Claimed limitation has the functionality corresponding to Claim 5. See rationale in Claim 5 above.

As per Claim 16: Regarding claimed limitation: *The method of claim 13, further comprising the step of forming a lock structure and a lock structure reference in an object header of the object, the lock structure including a lock holder and wait queues*: Claimed limitation has the functionality corresponding to Claim 7. See rationale in Claim 7 above.

As per Claim 17: Regarding claimed limitation: *The method of claim 12, wherein the step of determining to set the heavy lock includes determining that an address difference between a current lock slot of the first thread for the lightly locked object and that of the locking attempt by the second thread is greater than the reserved area*: Claimed limitation has the functionality corresponding to Claim 6. See rationale in Claim 6 above.

As per Claim 18: Regarding claimed limitation: *The method of claim 11, further comprising the step of addressing stack variables to identify threads using a stack-based local lock slot structure*: Claimed limitation has the functionality corresponding to Claim 8. See rationale in Claim 8 above.

As per Claim 19: Regarding claimed limitation: *The method of claim 18, wherein the step of determining to maintain the light object lock includes determining that an address difference between a current lock slot of the first thread for the lightly locked object and that of the nested intra-thread locking attempt is less than the reserved area*: Claimed limitation has the functionality corresponding to Claim 9. See rationale in Claim 9 above.

As per Claim 20: Regarding claimed limitation: *The method of claim 18, further comprising the step of forming a lock structure and a lock structure reference in an object header of the object, the lock structure including a lock holder and wait queues*: Claimed limitation has the functionality corresponding to Claim 7. See rationale in Claim 7 above.

Conclusion

4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Onodera, et al., "A study of locking objects with bimodal fields", ACM, discloses implementation of bimodal fields in Object locking.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (703) 308-9049. The examiner can normally be reached on Monday-Friday from 8:00 AM to 5:30 PM ET. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam, can be reached on (703) 305-4552.

The fax phone numbers:

(703) 872-9306 (for formal communication intended for entry);

(703) 746-5429 (for informal or draft communication, please label "PROPOSED" or "DRAFT").

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.

TED T. VO

Patent Examiner
Art Unit: 2122
June 10, 2004